

## CLAIMS

What is claimed is:

1. A system that facilitates message processing, comprising:  
an input component that receives a request to process a message; and  
a group management component that associates the message with related messages and automatically locks the message and the related messages *via* a conversation group identifier, the lock provides a request initiator exclusive access to the locked messages for processing.
2. The system of claim 1, the request includes at least one of a message identifier and a conversation group identifier.
3. The system of claim 2, the message identifier is utilized to associate the requested message with the related messages and form a conversation group.
4. The system of claim 2, the conversation group identifier determines the message's conversation group.
5. The system of claim 1, the requested message and the related messages are stored in-order in a message queue.
6. The system of claim 5, in-order storage is ensured by at least one of storing messages received out-of-order in temporary storage until they become in-order and dropping messages received out-of-order.
7. The system of claim 1, the requested message and the related messages within a conversation group are serially processed from a message queue.
8. The system of claim 1, an incoming related message is stored, dynamically associated with the conversation group identifier, and made available to the request initiator for processing.

9. The system of claim 1, the requested message and the related messages are associated with a plurality of related dialogs.
10. The system of claim 1, the requested message and the related messages are concurrently queued and dequeued.
11. The system of claim 1, the lock prevents another message processor from processing the related messages.
12. A system that facilitates access to application data, comprising:
  - a component that associates an application with a conversation group identifier;
  - and
  - a conversation management component that locks the application and associated application data *via* the conversation group identifier when the application is invoked, the lock provides exclusive access to the application data.
13. The system of claim 12, the component associates at least one other application with the conversation group identifier.
14. The system of claim 13, the other application and associated application data are locked concurrently with the invoked application.
15. The system of claim 12, the application data is serially processed.
16. The system of claim 12, subsequently stored application data is dynamically associated with the conversation group identifier and available for processing.

17. A method that facilitates message processing, comprising:  
requesting one or more messages;  
locking a conversation group that comprises at least the one or more messages,  
the lock prevents a disparate requestor from accessing the messages; and  
providing exclusive serial access to the messages.
18. The method of claim 17, further comprising associating related messages with the conversation group.
19. The method of claim 17, further comprising automatically updating the conversation group with incoming related messages.
20. The method of claim 17, further comprising utilizing a conversation group identifier to lock the conversation group.
21. The method of claim 17, further comprising storing the messages in-order.
22. A method that facilitates message processing, comprising:  
receiving a request for a message;  
associating the message and related messages with a conversation group;  
locking the conversation group *via* a conversation group identifier; and  
providing exclusive access to the conversation group.
23. The method of claim 22, further comprising utilizing message identifiers to determine whether messages are related.
24. The method of claim 22, further comprising receiving the conversation group identifiers to associate with the conversation group.
25. The method of claim 22, further comprising dynamically updating the conversation group with incoming related messages.

26. A method that facilitates serial access to application data, comprising:
  - associating an application with a conversation group identifier;
  - invoking the application;
  - locking the application *via* the conversation group identifier; and
  - providing exclusive access to application data.
27. The method of claim 26, further comprising utilizing an API to associate the application with the conversation group identifier.
28. A data packet transmitted between two or more computer components that facilitates message processing, comprising:
  - requesting a message for processing; associating the message with related messages in a conversation group; locking the conversation group *via* a conversation group identifier; and providing exclusive serial access to the requested and the related messages for processing.
29. A computer readable medium storing computer executable components that facilitate message processing, comprising:
  - a component that receives a request for a messages;
  - a component that automatically associates the requested message with related messages in a conversation group;
  - a component that locks the conversation group *via* a conversation group identifier;
  - and
  - a component that provides exclusive serial access to the messages within the locked conversation group.
30. A system that facilitates message processing, comprising:
  - means for associating a requested message with related messages;
  - means for locking the requested and related messages; and
  - means for serially accessing the requested and related messages.